Active Learning using Dirichlet Processes for Rare Class Discovery and Classification

Tom S. F. Haines
thaines@eecs.qmul.ac.uk
Tao Xiang
txiang@eecs.qmul.ac.uk

School of Electrical Engineering and Computer Science
Queen Mary, University of London
London, UK

Abstract

Real-world classification problems, such as visual surveillance and network intrusion detection, often contain common yet uninteresting background classes and rare but interesting classes, that need to be both discovered and classified. Active learning offers a suitable solution to joint rare class discovery and classification, by minimising the manual labelling of training data. A novel active learning approach is proposed, which automatically balances the competing goals of new class discovery and improving classification. Crucially it is free of tuneable parameters. Using Dirichlet processes a new active learning criterion is formulated, based on first computing the probability that unlabelled exemplars are from a new class, in addition to existing classes, and subsequently the probability of misclassification, which is then used for query selection. The proposed approach works with any probabilistic classification model and its effectiveness is demonstrated on multiple problems.

1 Introduction

Classification is a classic problem, and one that is well rehearsed. In many real-world problems, such as visual surveillance, computer network intrusion detection and financial fraud pattern detection, the proportion of exemplars in different classes is highly imbalanced - the majority of the examples belong to uninteresting background classes whilst the interesting classes have few examples. For example in the Sloan Digital Sky Survey most of the survey images of galaxies and quasars capture known phenomena, whilst unusual phenomena that could be evidence of new science constitute only 0.001% of the total dataset [13]. Crucially, those interesting rare classes are often not known a priori and need to be discovered. To both discover and classify rare classes one typically needs to exhaustively label the entire dataset, to obtain sufficient instances of each rare class. Such a manual labelling process is often prohibitively expensive, rendering a supervised learning approach impractical.

Active learning offers a solution by minimising the manual labelling requirement. There are two competing goals - to find all the rare classes, and to refine the boundaries between the known classes. However, most existing active learning methods either assume that all classes are known and thus focus on the classification problem [15], or focus on the class discovery problem only [8, 13, 21]. The approaches that try to meet both goals simultaneously [10, 18] are heuristic, and have free parameters that need tuning for each scenario.

A novel active learning approach is proposed, which automatically balances the two competing goals and has no free parameters to tune. It takes the standard form of iteratively selecting a problem *instance* from a *pool* of instances. For each iteration the selected instance is given to the *oracle*, which provides the class label, and then used to update the model. The selection proceeds in three steps. Firstly, for each instance in the pool the probabilities of it belonging to each existing class, and also the probability of it belonging to a new class, are calculated under a Dirichlet process (DP) assumption. Secondly, the probability that the instance will be misclassified is calculated. Normally this would be an *uncertainty* method of active learning, and would only improve the boundary between existing classes, but by considering the possibility of the instance belonging to a new class when the classifier can only assign known classes this metric also achieves the goal of class discovery. The balance between the two goals is decided by the concentration parameter of the DP, which is automatically inferred. Finally, a single instance is selected, based on the estimated chances of misclassification. The key contribution is this novel active learning criterion, which is specifically designed to balance the two competing goals of discovery and classification. Furthermore, the simple implementation, lack of tuneable parameters and model-agnostic design makes this approach particularly attractive[1].

Related work: Active learning is a long standing [1] and expansive field - the survey of Settles [15] gives an overview. The basic objective is to improve on random sampling[2]. Various active learning approaches differ in the criteria adopted, which broadly fall into the two categories of *likelihood* and *uncertainty* based sampling. The likelihood criterion [9] proceeds by querying the exemplars that have the lowest probability according to the classifier's current model. It is better suited to finding new classes than refining the boundaries of existing classes. Likelihood is limited however by its inability to distinguish new classes from outliers, and to find classes that are inseparable from already detected classes. The uncertainty criterion selects instances for which the classifier is uncertain - it is thus good at refining the boundaries between classes, but not at class discovery. Multiple uncertainty methods exist [15], including the entropy method and the *query by committee* method [16].

Most existing active learning studies assume that all classes are known a priori. Recently there have been a number of works that focus on the rare class discovery problem. Pelleg & Moore [13] use an EM classifier with Gaussian distributions and adopt a variant of the likelihood criterion. Whilst it is specifically for finding rare classes the total number of rare classes must be provided up front to set the number of EM clusters. The model and active learning method are also inseparable. He & Carbonell [8] perform density estimation and query exemplars based on identifying local maxima in the density using gradients. Like [13] this requires knowing how many unknown classes exist; additionally it also needs the expected ratios for the classes - for real problems this is unreasonable. Vatturi & Wong [21] also take a density estimation approach. Mean shift at multiple scales is used to cluster the examples in the pool, and for each cluster the example nearest to the centre is selected to be queried. It gives strong performance for rare class discovery. However, both [21] and [8] avoid interacting with the classification model. This is advantageous as it applies no restriction on the model, but problematic as it can only work to find new classes, not to improve the classification model, so poor results are expected for classification.

The closest work is that of Hospedales et al. [10], which also tries to balance the competing goals of class discovery and boundary refinement when dealing with rare classes. In their

[1] An implementation may be obtained from http://thaines.com

[2] Sometimes referred to as *passive learning*.

approach two models and two active learning criteria are considered simultaneously, in a switching framework. They are a generative model (kernel density estimate) with likelihood detection, and a discriminative model (support vector machine) with uncertainty detection. As the querying progresses it switches between the models based on their past performance. This initially means it mostly uses the generative model, which is better when given few examples, to find new classes, but latter tends to be discriminative, to refine the boundary between classes. Not surprisingly, it outperforms previous active learning methods which are designed for solving either class discovery or classification, not both. However, the method is entirely heuristic and includes parameters that need to be tuned for each scenario.

2 Methodology

Pool based learning [15] is presented, though the algorithm could be readily adapted for a stream based approach. Given a pool of problem instances the algorithm consists of a loop where first a specific instance is selected from the pool, secondly the instance is fed to the oracle and labelled, and thirdly the model is updated with the new labelled instance. To determine which instance to query at each iteration of the loop for each instance a distribution is computed, that represents the probability of the instance belonging to each existing class, and, additionally, belonging to a previously unseen class. Given these probabilities the probability of misclassification for each instance in the pool given the current model can be computed. These two tasks are detailed in the following two subsections, with an illustrative demonstration given in the third subsection.

2.1 New class probability

Calculating the probability that an instance comes from an unknown class is problematic, as, by definition, nothing is currently known about the class. To resolve this it is assumed that the generating procedure can be modelled using a generative model with a *Dirichlet process* (DP) as the prior. This is a valid assumption for most classification problems [14].

A Dirichlet process [4] is typically used for non-parametric Bayesian models, e.g. density estimates [3] and topic models [20]. It can be formally described as a Dirichlet distribution with the number of components taken to infinity, or using a construct called the stick-breaking process [14]. For the purpose of active learning however two properties are important: that it has clustering behaviour [19], such that it expects the instances to be grouped into discrete classes; and that it considers an infinite number of classes, and hence dynamically adjusts the number of classes given the data.

The Dirichlet process may be denoted as $DP(\alpha, \beta)$, where α is its *concentration parameter* and β is its *base measure*. Its marginal posterior, the Chinese restaurant process [2], is all that needs to be calculated for active learning. The Chinese restaurant process is an analogy consisting of a restaurant containing an infinite number of tables at which customers sit, where each table is in effect a cluster of patrons. On each table only a single dish is served, representing a single choice from the menu. This represents a draw from the base measure. When new customers arrive they either sit at a table with existing patrons, and consume the dish already assigned to the table, or they choose a previously unused table and a new dish from the menu (base measure) is selected (drawn) for the new table. These correspond to the instance belonging to an existing class and a new class, respectively. Each of the in-use tables is chosen proportional to the number of customers already sitting at them, whilst a

new table is chosen proportional to the concentration parameter. Note that whilst an infinite number of tables theoretically exist, corresponding to the components of the infinitely sized Dirichlet distribution, only used tables need to be tracked, making this a finite construction.

For each instance in the pool of unlabelled instances the aim is to compute the probability of it belonging to each existing class, and of it belonging to a new class, conditional on all previous instances for which the oracle has provided a label. This is assuming a mixture-like model, where each table in the DP corresponds with a class assignment. Note that this is not a requirement for the classification model to also be a mixture model; it can be any probabilistic model where $P_c(\text{data}|\text{class})$ can be calculated. For the moment the existence of a prior, $P(data)$, is also assumed, such that $P_c(\text{data}|\text{class})$ is its posterior, using Bayes rule. Accordingly, the probability distribution for an instance is given as

$$P_n(c \in C \cup \{\text{new}\}|d) \propto \begin{cases} \frac{m_c}{\sum_{k \in C} m_k + \alpha} P_c(d|c) & \text{if } c \in C \\ \frac{\alpha}{\sum_{k \in C} m_k + \alpha} P(d) & \text{if } c = \text{new} \end{cases} \tag{1}$$

where d is the data for the considered instance, C is the set of known classes, m_c the number of instances labelled with class c and α is the concentration parameter for the DP. Once normalised this provides a distribution for each instance that consists of the probability of the instance belonging to each of the known classes as well as to an unknown class. Two issues remain - how to set the concentration parameter and how to set the prior, $P(\text{data})$.

Instead of treating α as a user set parameter a prior may be applied and Gibbs sampling used to estimate it, using the technique of Escobar & West [3]. The prior on α is a gamma distribution, $G(a,b)$. This method proceeds by first sampling a quantity η given the current concentration, and then resampling the concentration given η. η given the concentration, α, is given in terms of the beta distribution, $B(.,.)$

$$\eta|\alpha,k,n \sim B(\alpha+1,n) \tag{2}$$

where k is the number of classes that currently exist and n the number of examples distributed over the classes. α given η is then a mixture of two gamma distributions, $G(.,.)$

$$\alpha|\eta,k,n \sim \pi G(a+k,b-\log(\eta)) + (1-\pi)G(a+k-1,b-\log(\eta)) \tag{3}$$

where the ratio of the mixing terms is given by

$$\frac{\pi}{1-\pi} = \frac{a+k-1}{n(b-\log(\eta))} \tag{4}$$

Given a prior the mean of a number of Gibbs samples is used, after a burn in period. In this work a weakly-informative prior of $G(1,1)$ is used, with 128 samples used for both burn in and sampling the mean; for initialisation the concentration of the previous query is used.

A prior, $P(\text{data})$, is also required. Whilst a proper prior can certainly be used this term obviously parallels active learning methods based on density estimation (Such as [21]) - it defines how likely a sample is something useful, rather than an outlier. It follows that the prior must be selected based on the data in the pool, for which it is in effect going to be a density estimate. Given that real priors are often very simple, e.g. conjugate, a good density estimate will be beneficial, and as there is no reason to use an actual prior a proper density estimate based on the initial pool is preferred.

2.2 Misclassification probability

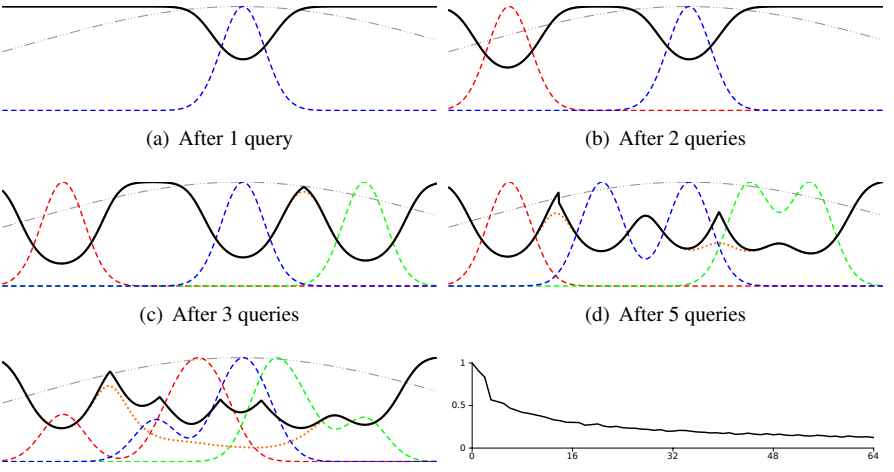
Given the class membership probabilities, $P_n(.)$, which include the probability of belonging to a new class, an actual selection from the pool is required. The goal is to balance finding new classes against refining existing classes. A common approach to improving the existing model is to select instances that have a high degree of uncertainty in their classification given the current model. The most popular method is the entropy method, but entropy cannot be applied when there is a probability of an unknown class, at least not without the introduction of free parameters. Several alternative approaches to entropy exist [15]. One such approach is to calculate the probability of classifying an instance incorrectly. For instance this approach was implicitly used by Lewis & Gale [12] for the purpose of text classification. They described it in terms of selecting instances with class probabilities that are closest to 0.5, which is equivalent. To include the possibility of a new class this idea has to be considered explicitly, and proper consideration of multiple classes has to be made.

Two assumptions are made - firstly that the classifier will select the class to which it has assigned the highest probability, noting that this only includes known classes, and secondly that the calculated distribution is an accurate estimate of what the true class of the instance could be, noting that it includes the possibility of a new class. It is then a simple matter to calculate the probability of incorrectly classifying an instance,

$$P(\text{wrong}|\text{data}) = 1 - P_n(c|\text{data}), \qquad c = \underset{c \in C}{\operatorname{argmax}} P_c(c|\text{data}) \tag{5}$$

where $P_n(c|\text{data})$ is the probability of the instance belonging to the selected class as calculated above, whilst $P_c(c|\text{data})$ is the probability calculated by the classifier, typically using Bayes rule with a $P(c)$ term. If $P(c)$ weights classes by the number of instances seen then $P_n(c|\text{data})$ and $P_c(c|\text{data})$ will be equivalent, other than P_c excluding the probability of a new class and hence being normalised differently. Alternatively, if a different prior on class probability is assumed, e.g. a uniform distribution, then this will not be the case. Finally, to select an instance from the pool $P(\text{wrong}|\text{data})$ is used, to weight each instance for a weighted draw.

It is important to note that the proposed misclassification probability (denoted as $P(\text{wrong})$ hereafter) based active learning criterion is different from a conventional uncertainty criterion that focuses only on boundary refinement for existing classes. This is because $P_n(c|\text{data})$ includes the probability of the instance belonging to a new class (denoted as $P(\text{new})$), which the classifier can never select. If the $P(\text{new})$ value is high, the $P(\text{wrong})$ value will also be high; similarly if the $P(\text{new})$ value is low but the classifier is uncertain, so that none of the class probabilities are high, a high $P(\text{wrong})$ will again be generated. Therefore the value of $P(\text{wrong})$ is determined by two factors: the likelihood that the instance belongs to an unknown class, and how uncertain the current classifier is about the instance. Which of these two dominates is driven by the concentration parameter. Specifically, when it is high relative to the number of labelled instances selection becomes equivalent to using $P(\text{new})$ directly, but as it heads to zero only classification uncertainty is considered, and the boundaries are refined. In practice the concentration parameter relative to the instance count tends to start high and drop to a low but constant level as the number of queries increases, i.e. as expected it starts by finding new classes, but as it sees more data and stops finding them it refocuses on boundary refinement.



(a) After 1 query (b) After 2 queries

(c) After 3 queries (d) After 5 queries

(e) After 12 queries (f) Interest in new class discovery

Figure 1: 1*D* demonstration of the problem with 3 classes, with probability distributions. The prior is constant, indicated by the dot-dash grey line, whilst the distributions for the 3 classes use the 3 primary colours, dashed. Orange dots are used for the $P(\text{new})$ metric, whilst $P(\text{wrong})$ is given in black.

2.3 An illustrative demonstration

How the algorithm works is visualised using a 1*D* problem. Specifically, the 4*D* 3-class *iris* problem of Fisher [5] is used, as obtained from the UCI repository [6]. It is a classification problem where the task is to identify flower species based on flower shape measurements. Principal component analysis (PCA) is used to reduce the problem to a single dimension[3]. Many approaches can be selected for classification. For this and the other experiments presented in the paper the incremental kernel density estimation (KDE) method of Sillito & Fisher [17] is used. It uses a Gaussian kernel whilst the number of mixture components is capped, to maintain a constant time incremental algorithm. When the cap is passed mixture components are optimally merged. One density estimate over the pool is used as the pseudo-prior, whilst each class also has a density estimate built from its members. A uniform prior over class assignment is used and kernel size is selected using leave one out cross-validation.

Figures 1(a) through to 1(e) show the state of the system after the given number of queries. They are plots of probability values calculated for every 1*D* feature vector, where each plot has been separately normalised to fill the available height. Firstly, the prior is given by the grey line - it does not change as the system runs. Each of the known classes is coloured using a primary colour. The $P(\text{new})$ curve, giving the probability that a point on the line is going to belong to a new class, is given in orange whilst the $P(\text{wrong})$ curve is given in black. $P(\text{new})$ has been included as it makes the behaviour of $P(\text{wrong})$ with regards to boundaries clearer. The $P(\text{wrong})$ graph indicates how interested in a point the presented algorithm is, with the high points being the positions the algorithm is likely to select for its next query, conditional on such locations actually appearing in the data set of course. Figure 1(f) plots the concentration normalised by the concentration plus the number of labelled instances, i.e.

[3]It is not really solvable after this, as the classes have a lot of overlap, but is sufficient to illustrate the inner workings of the presented approach, whilst the reduction to 1*D* allows for a clean visualisation.

	shuttle		gait		digits	
	discovery	classification	discovery	classification	discovery	classification
random	486.2	53.5	1170.5	78.9	915.2	54.6
entropy	423.5	51.8	1183.8	75.3	974.0	57.1
likelihood	950.5	79.4	1171.7	56.5	1060.2	61.9
Pelleg [13]	534.0					
He [8]	768.5					
Vatturi [21]	970.5					
Hospedales [10]	933.2	61.8	1253.1	84.8	1207.4	69.5
P(wrong)	923.4	79.8	1241.9	88.4	1133.6	69.7

Table 1: Areas under the various graphs, for the first 150 queries, with the highest score in each column highlighted.

the weight assigned to new classes given the number of queries made.

Firstly, a new class is found in each of the first 3 queries, and with the weight assigned to finding new classes dominating the two methods have identical interests (the orange line is underneath the black line). After the third query a slight difference is evident in that P(wrong) is more interested in examples that are on the boundary between the blue and green classes. The state after 5 queries demonstrates that, as the algorithm loses interest in finding new classes, as plotted in figure 1(f), the two approaches start to differ, with P(wrong) showing greater interest in the classification boundaries whilst still maintaining an interest in areas where new classes could be. By 12 queries this is much more pronounced. This demonstration clearly shows the various behaviours expected - an interest in areas where either new classes could be or the boundary could be refined, with the latter gaining dominance as it loses interest in finding new classes. Figure 1(f) demonstrates how the level of interest in finding new classes drops as the algorithm makes more queries[4].

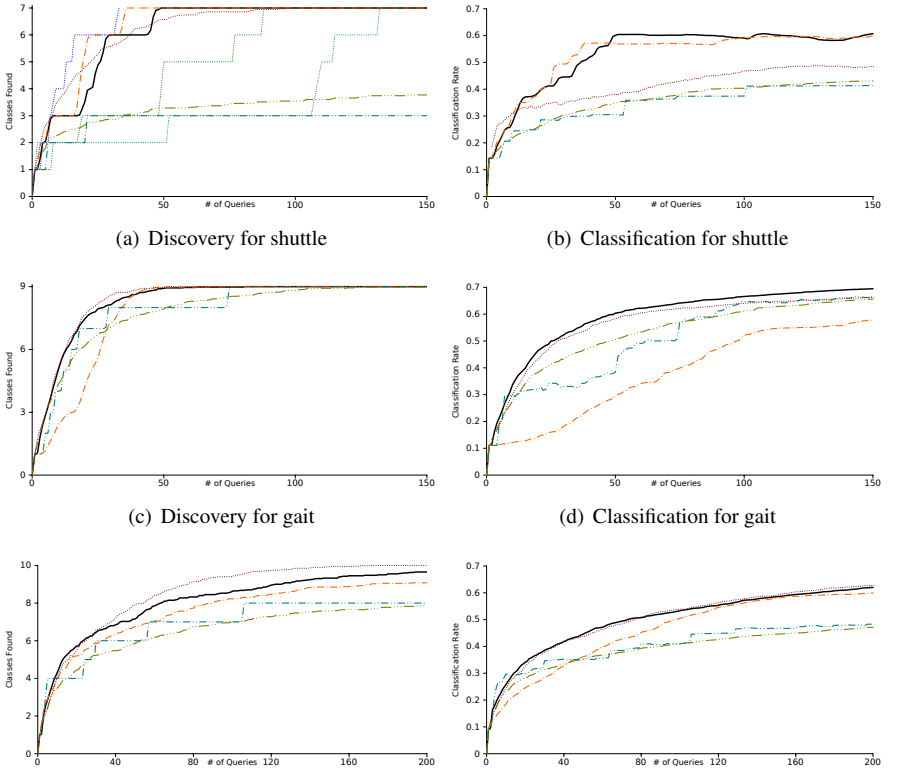
3 Experiments

Three datasets are demonstrated. The *shuttle* dataset is part of the UCI repository [6]. It has 7 classes, which are naturally unbalanced, with 78% of instances in the largest class and 0.01% in the smallest class. The next two datasets are *gait* [22] and *digits* [11]. Gait consists of videos of various people walking, recorded from 9 different angles - the problem is to classify the viewing angle given the *gait energy image* [7] of the walker[5]. Digits consists of the 10 handwritten digits, the goal being to recognise which digit is represented. Figure 3 contains examples of the images from both datasets. In both cases the original datasets are balanced. They are subsampled so that the class sizes follow a geometric progression. For gait this is done so there are 200 members in the largest class and 12 members in the smallest, whilst for digits it is 4096 members in the largest class and 8 in the smallest. Principal component analysis is used in both cases to reduce the number of dimensions to 25.

We follow the same testing procedure as Hospedales et al. [10] for the purposes of comparison. Specifically it consists of letting the algorithm make its queries from a pool of training instances, and recording two quantities after each query - how many classes have been found (discovery) and the classification accuracy on a separate test set (classification).

[4]Note that concentration cannot be calculated until at least two classes have been found, hence the jump in the graph at that time.

[5]This is the silhouette of the walker averaged over multiple frames, whilst being tracked and aligned such that overlap between frames is maximised.



(a) Discovery for shuttle (b) Classification for shuttle

(c) Discovery for gait (d) Classification for gait

(e) Discovery for digits (f) Classification for digits

Figure 2: Graphs showing algorithm results versus the number of queries made to the oracle. On the left the average number of classes found, on the right the mean classification accuracy. Each row corresponds to a dataset - shuttle, gait and digits as labelled. The legend is to the right, noting that not all algorithms are in all graphs.

Both of these metrics are plotted for all datasets in figure 2, whilst the areas under the graphs are given in table 3. In all cases the averages of multiple runs are used. For classification the previously detailed KDE method [17] is used, which is the same generative model used by Hospedales et al. [10]. A variety of algorithms are compared, including *random*, which is the baseline approach of selecting queries at random; *entropy*, which is the entropy-based uncertainty method; and *likelihood*. The algorithms of 4 competitors are included, as available, and are labelled by the first author of their respective papers [8, 10, 13, 21].

For the *shuttle* dataset, on class discovery Vatturi [21] is the clear winner, but their approach is dedicated to finding rare classes, not improving classification performance. Given that it does not consider the classification method, choosing an order in which to select instances before training ever starts, it is not expected to be competitive for classification. The likelihood method does surprisingly well, and is the second best at discovery, a behaviour not seen in the other datasets. This can probably be attributed to the nature of the dataset, which has little noise and good seperation, giving an outlier detector a significant advantage for class discovery. It is hardly surprising that finding classes quickly can result in good

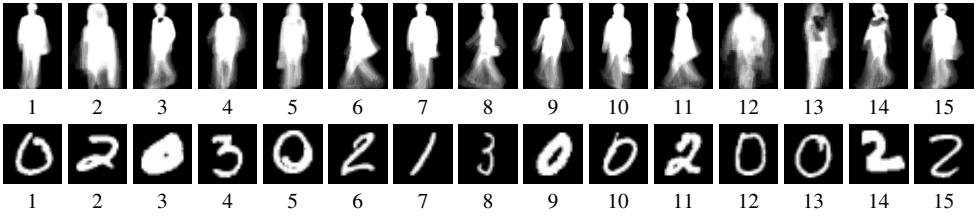
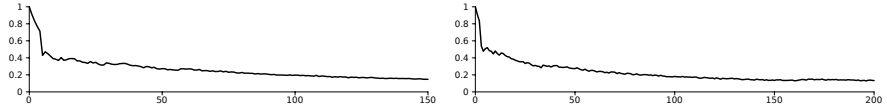


Figure 3: The selected problem instances for the first 15 queries made for both the gait (top row) and digits (bottom row) datasets - the query number is beneath each image.



(a) gait (b) digits

Figure 4: Plots of the inferred concentration value normalised by the concentration plus the number of instances that have already been labelled. It reflects how much weight the algorithm assigns to finding new classes.

classification performance, which likelihood also demonstrates. The presented approach is effectively joint best with likelihood for classification performance.

For the two vision datasets, gait and digits, the results of [8, 13, 21] are not available. It can be seen from figure 2 and table 3 that in both cases Hospedales [10] takes first place for discovery, with the proposed algorithm ($P(\text{wrong})$) coming in second, whilst for classification our algorithm is slightly better. The sequences of images queried by $P(\text{wrong})$ for both datasets are given in figure 3. It can be observed that whilst it repeatedly requests instances from the same class, each query is different from the other, e.g. in the digits sequence 7 zeros are queried, every one of them representing a markedly different way of writing it. Additionally, figure 3 gives just how interested $P(\text{wrong})$ is in finding new classes against the number of queries, demonstrating how it tends to go down with time, such that it focuses on improving class boundaries once it stops finding new classes.

As demonstrated by the results the proposed algorithm slightly outperforms Hospedales [10] on classification, the metric that both approaches attempt to maximise by balancing discovery with boundary refinement. Both of them give significantly better performance than most existing algorithms, that focus on a single goal. It is hence interesting to note the differences between the two models. Firstly, and key to our argument, is that Hospedales is heuristic, and has parameters that need tuning for each specific problem, whilst $P(\text{wrong})$ has no parameters and gets the performance demonstrated without any kind of tuning - this is highly advantageous in the real world, where parameter tuning can be difficult, sometimes impossible. Additionally, Hospedales has two classifiers available - a generative one, also used for the presented experiments with $P(\text{wrong})$, and a support vector machine (SVM). It can be observed that whilst for the first couple hundred queries $P(\text{wrong})$ can do better, in the long run Hospedales will always get the better classification performance, due to the use of a SVM, which is better than a generative model given sufficient data. Given that Hospedales uses a better classifier and relies on parameter tuning we would argue that the proposed is clearly the better approach, especially given the simpler implementation of $P(\text{wrong})$.

4 Conclusions

A simple active learning method free of tuneable parameters has been demonstrated to have better performance than the conventional approaches, and equivalent or slightly better performance than a complex heuristic model that requires parameter tuning for each dataset. It is a practical algorithm that can be implemented quickly and used without modification regardless of the data, and will work with any classifier that can provide probabilistic output.

Two issues for further work can be raised. The first is that the DP assumes a logarithmic relationship between the number of samples drawn and the number of classes - if the data varies greatly from this assumption then it could be suboptimal. An alternative distribution could be considered, such as the Pitman-Yor distribution, which allows for power law behaviour, which is much more common with real data. The second is that the approach does not work with discriminative methods, such as SVMs, and a means to integrate such techniques would be ideal, as this is a disadvantage when lots of labelled instances are available.

References

[1] D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[2] D. Blackwell and J. B. MacQueen. Ferguson distributions via Polya urn schemes. *Annals of Statistics*, 1(2):353–355, 1973.

[3] M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *J. American Statistical Association*, 90(430):577–588, 1995.

[4] T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.

[5] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[6] A. Frank and A. Asuncion. UCI machine learning repository, 2010. URL http://archive.ics.uci.edu/ml.

[7] J. Han and B. Bhanu. Individual recognition using gait energy image. *Pattern Analysis and Machine Intelligence*, 28(2):316–322, 2006.

[8] J. He and J. G. Carbonell. Nearest-neighbor-based active learning for rare category detection. *Neural Information Processing Systems*, 21, 2007.

[9] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[10] T. M. Hospedales, S. Gong, and T. Xiang. Finding rare classes: Adapting generative and discriminative models in active learning. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 15, 2011.

[11] Y. LeCun and C. Cortes. Mnist database of handwritten digits. URL http://yann.lecun.com/exdb/mnist/.

[12] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. *Proc. Conf. on Research and Development in Information Retrieval*, 17:3–12, 1994.

[13] D. Pelleg and A. Moore. Active learning for anomaly and rare-category detection. *Advances in Neural Information Processing Systems*, 17:1073–1080, 2004.

[14] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[15] B. Settles. Active learning literature survey. Technical Report 1648, Uni. of Wisconsin-Madison, 2009.

[16] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. *Proc. Workshop on Computational Learning Theory*, 5:287–294, 1992.

[17] R. R. Sillito and R. B. Fisher. Incremental one-class learning with bounded computational complexity. *International Conference on Artificial Neural Networks*, 17:58–67, 2007.

[18] J. W. Stokes, J. C. Platt, J. Kravis, and M. Shilman. ALADIN: Active learning of anomalies to detect intrusion. Technical Report 2008-24, Microsoft Research, 2008.

[19] Y. W. Teh and M. I. Jordan. *Bayesian Nonparametrics*, chapter Hierarchical Bayesian Nonparametric Models with Applications. Cambridge University Press, 2010.

[20] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *J. American Statistical Association*, 101(476):1566–1581, 2006.

[21] P. Vatturi and W-K. Wong. Category detection using hierarchical mean shift. *Knowledge Discovery and Data mining*, 15:847–856, 2009.

[22] S. Zheng. Casia gait database. URL http://www.cbsr.ia.ac.cn/english/Gait%20Databases.asp.